

# Form の作成法

## バリアオプションにフォームを付ける

TAKEUCHI juichiro  
SHIMURA Masato

2009年5月26日

### 目次

1	FORM の概要	1
1.1	form editor を起動 . . . . .	1
2	部品	3
2.1	radiobutton . . . . .	3
2.2	button . . . . .	4
2.3	combodrop . . . . .	6
2.4	edit の作成 . . . . .	7
2.5	見出しの漢字英語切り替え . . . . .	7
2.6	ANS の表示 . . . . .	10
3	バリアオプションのシュミレーション	10
4	計算結果の記録と保存	10
5	Reference	11

\*1

### はじめに

#### 1 FORM の概要

##### 1.1 form editor を起動

---

\*1 SHIMURA jcd02773nifty.ne.jp

このような form を作成する。

ijs の画面 (開いていなければ新規で作成) で **Edit** から *form editor* を選ぶ

*form editor* の最初の画面 *form Id* が全体を統合する名  
type は *baseform.ijs* を選ぶ  
OK で次の 2 画面に切り替わる。

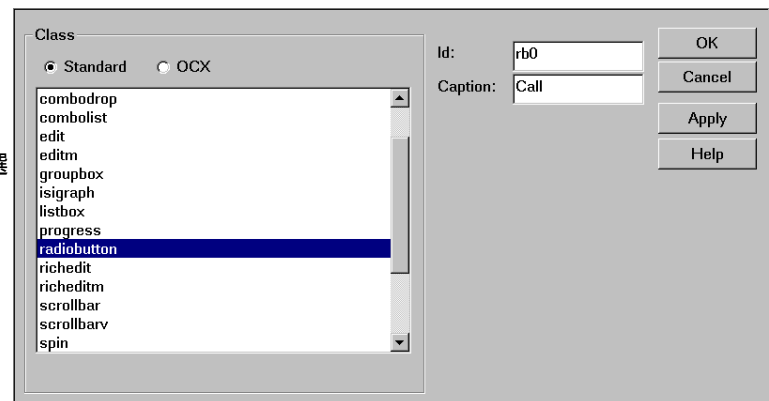
## 2 部品

よく使う次のフォームを作成する。種々のフォームの形態やスクリプトは  
system/examples/demo/control(1,2,3).ijs  
等を参照

- radiobutton
- button
- combogrop
- edit/editm

### 2.1 radiobutton

*Design* の方の画面で **New** を選ぶ。radiobuttonh を選  
び **Id** に一つずつ異なる名前を付ける。ここでは *rb0*  
OK で進む



このとき左上は後続のフォームの原型が出てくる場所なので 3cm ほど空けておき、最後に画面を調整すると良い。

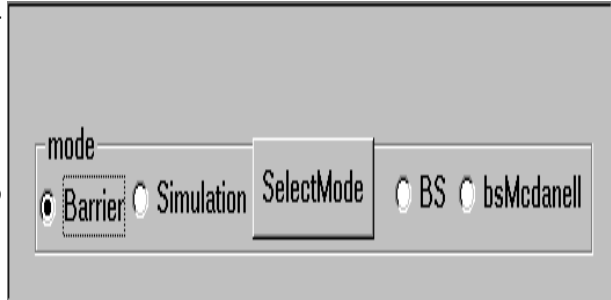
OK でスクリプトが自動で出力される。慣れれば座標などを手で編集しても良い。

```
xywh 14 61 38 11;cc rb0 radiobutton;cn "Call";  
xywh 55 61 35 11;cc rb1 radiobutton group;cn "Put";  
xywh 9 55 81 19;cc radio01 groupbox;cn "Call/Put";
```

- **groupbox** ラジオボタンはグループで囲む。
- 最初に **New** で **groupbox** を選ぶ。Id は *radio01*, caption は *Call/Put*。
- マウスでドラッグして好みの位置に持っていき、マウス右ドラッグで「右」と「下」が伸びるので整形する。
- **New** で *radiobutton* を選ぶ。2 個作成
- 最初の **Id** は *rb0*, **caption** は *Call* とする。(どちらも任意の名前を付ける)
- 任意の場所にマウスでドラッグする。この前に OK Cansel のボタンを下の方に移しておく
- **rb0** こちらが最初。(Call)
- **rb1** 2 個目のボタン (Put) **group** が入る
- *rb0* と *rb1* はグループ化され、択一式の選択となる。



- 同様にモード選択の radiobutton を 4 個作成する。(rb4-rb7)
- **groupbox** で囲む。
- このモード選択のボタンを作成し、選んだモードに応じて入力スペースの表示を変える。このボタンを認識すると OK ボタンより先にモード選択ができる。



```
xywh 10 22 33 11;cc rb4 radiobutton ;cn "Barrier";
xywh 42 21 43 11;cc rb5 radiobutton group;cn "Simulation";
xywh 133 20 22 11;cc rb6 radiobutton group;cn "BS";
xywh 155 20 45 11;cc rb7 radiobutton group;cn "bsMcdanell";
xywh 9 15 193 19;cc radio47 groupbox;cn "mode";
```

## 2.2 button

```
xywh 84 18 42 14;cc selsim button;cn "SelectMode";
```

- **New** で *button* を選び Id を *selsim*, Caption を *SelectMode* とする。
- ボタンをマウスドラッグでラジオボタンの脇に持ってくる。
- ボタンをクリックしたときの作用を定義する。

**radiobutton** ラジオボタンの情報を得る。

```
SEL=. ". rb4,rb5,rb6,rb7
```

*rb4* の初期値は 1 (文字) で, *rb4 - rb7* には 0 が文字で入っている。

```
1000 文字
```

数値化 (".) すると 4 個のボタンでは *SEL* は 10 進法で 1000,100,10,1 の値を取る。

\*2

ここで *rb5* をクリックしても *rb5* 値の変更がすぐには反映されないので次の関数を作成する。  
selsim button このボタンは *SelectMode* と表示されている。このボタンをクリックする動作を定義する。ボタンの名前は **Id** を用いる。

- *rb5* で simulation を選ぶと combobox(0) をセットして選択肢を表示する。
- *rb6,rb7* は BarrierPrice の入力窓（後出）を使わないので *notusing* と表示する。

```
BarrierOption_selsim_button=: 3 : 0
if. 100= SEL do. NB. select simulation
  set_combo0 ''
  set_combo1 ''
end.
if. SEL e. 10 1 do.
  wd 'set ed2 * ','not using'
  else. wd 'set ed2 * ',''
end.
)
```

```
menupop "File";
```

右の部分は標準で付いてくるが目障りならば消しても menu new "&New" "" "" "";

良い。 menu open "&Open" "" "" "";

ここを編集していろいろな機能を割り付ける方法も menusep;

ある。 menu exit "&Exit" "" "" "";

```
menupopz;
```

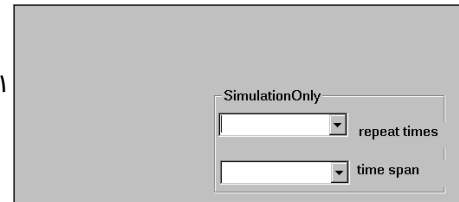
*ijs* を Ctrl/W で *ijx* に送り、*barrieroptiont\_run''* で実際の画面が出る。

## 2.3 combodrop

### 2.3.1 combodrop の作成

*selectmode* の選択に従いリストを **combodrop** に出す。

- 先ず **combodrop** を作成する。
- **New** で combodrop を作成する
- マウスで移動する。枠が赤いときに枠内をクリックしていると下の画面が出てくるので *vscroll* をクリックする。後に手で *ws\_vscroll* と書いても良い。  
`xy..;cc combodrop ws_vscroll;`



\*3

### 2.3.2 選択リストの登録

- 登録する List を作成する。

```
set_combo0=: 3 : 0
TIME0=. ": L:0 {@>1000 10000 100000
wd 'set combo0 ',; TIME0 ,each LF
wd 'setselect combo0 0;'
)
```

```
set_combo1=: 3 : 0
TIME1=. ;:'OneWeek ThreeDay OneDay'
wd 'set combo1 ',;TIME1 , each LF
wd 'setselect combo1 0;'
)
```

- シーケンシャルマシーン (;:) を使用する
- ,each LF で後ろにスペースを入れる
- 登録する。 *setselect* で最初に出る項を指定する。

### 2.3.3 combodrop の見出し

見出しを *static* で付ける

```
xywh 159 49 41 11;cc st6 static;cn "repeat times";
xywh 158 65 44 11;cc st7 static;cn "time span";
```

---

\*3 combo 系が 3 ある。listbox もほぼ同様の手法である。

## 2.4 edit の作成

画面からパラメーターを入力する *edit* を作成する。

見出し (static) は日本語と英語表示を切り替えるので省略する

```
xywh 3 91 60 12;cc ed0 edit;  
xywh 70 91 60 12;cc ed1 edit;  
xywh 135 91 60 12;cc ed2 edit;  
xywh 5 125 60 14;cc ed3 edit;  
xywh 70 126 60 12;cc ed4 edit;  
xywh 135 126 60 12;cc ed5 edit;
```



## 2.5 見出しの漢字英語切り替え

WIN では XP 以降で UNICODE を完全に扱うことができる。WIN2000 では表示されない。J は J5 以降。

- static を 6 個作成する。caption は空白にしておく。

```
xywh 4 78 60 11;cc st0 static;  
xywh 70 78 60 11;cc st1 static;  
xywh 135 79 60 11;cc st2 static;  
xywh 0 111 72 11;cc st3 static;  
xywh 75 110 54 11;cc st4 static;  
xywh 134 110 60 11;cc st5 static;
```

- 文字を登録する

```
NB. -----ENGLISH_JAPAN-----  
c_selek=: 'English'  
c_st0=: 'スポット価格'  
c_st1=: '行使価格'  
c_st2=: 'バリア価格'  
c_st3=: 'オプション期間(月)'  
c_st4=: 'ボラティリティ (%)'  
c_st5=: 'リスクフリー金利 (%年)'  
  
e_selek=: '漢字'
```

```
e_st0=: 'CurrentPrice'  
e_st1=: 'StrikePrice'  
e_st2=: 'BarrierPrice'  
e_st3=: 'TimeInYears(Month)'  
e_st4=: 'Volatility(%) year'  
e_st5=: 'InterestRate(%) year'
```

- 英語表記

```
ek_set_eng=: 3 : 0  
NB. initial set  
wd'setfont selek ',font  
wd'setfont st0 ',font  
wd'setfont st1 ',font  
wd'setfont st2 ',font  
wd'setfont st3 ',font  
wd'setfont st4 ',font  
wd'setfont st5 ',font  
NB. -----  
wd'setcaption selek *', '漢字' NB. button  
wd'set st0 *',e_st0 NB. editm  
wd'set st1 *',e_st1  
wd'set st2 *',e_st2  
wd'set st3 *',e_st3  
wd'set st4 *',e_st4  
wd'set st5 *',e_st5  
btextCheck=: 1 NB. display button [KANJI]  
)
```

- 漢字表記 漢字はijsで直接入力できないときはnotepadなどUNICODE対応のエディタで書いて張り付ける。

```
ek_set_jp=: 3 : 0  
wd'setfont selek ',font  
wd'setfont st0 ',font  
wd'setfont st1 ',font  
wd'setfont st2 ',font  
wd'setfont st3 ',font  
wd'setfont st4 ',font  
wd'setfont st5 ',font  
NB. -----  
wd'setcaption selek *', 'ENGLISH'
```



```

wd'set st0 *',c_st0 NB. editm
wd'set st1 *',c_st1
wd'set st2 *',c_st2
wd'set st3 *',c_st3
wd'set st4 *',c_st4
wd'set st5 *',c_st5
btextCheck=: 0 NB. display button [English]
)

```

### 2.5.1 切り替えボタン

- 切り替えボタンを *button* で作成する。  
xywh 6 4 48 12;cc selek button;
- ボタンの表示は *caption* ではなく、次により行う。  
wd'setcaption selek \*', '漢字' NB. button
- ボタンによる切り替え *btextCheck* を監視する。ボタンを押したら値を変更する。

```

btextCheck=: 1/0 NB. display button [KANJI]

```

```

BarrierOption_selek_button=: 3 : 0
select. btextCheck
  case. 0 do. NB. push Kanji
    ek_set_eng ''
  case. 1 do. NB. push English
    ek_set_jp ''
end.
)

```

## 2.6 ANS の表示

FORM から計算すると答えは `ijx` にはもう出ないので答えを表記するスペースを設ける。

```
xywh 8 145 123 23;cc ans static rightmove bottomscaleg;
```

## 3 バリアオプションのシュミレーション

竹内寿一郎「エキゾチックオプション価格のシュミレーション (その 11) バリアオプションの価格式 (6)」  
JAPLA2009/04 によった。

オプション期間の値は月に統一した。

比較のためブラックショールズモデル (M.Shimura , E Mcdnald) を載せてみた。

Barrier Call	120 100 90 6 30 5 Call=24.1793	1 1 Barrier 120 100 90 6 30 5  24.1793
simuration Call 1000 Oneweek	120 100 90 6 30 5 Call=24.2578	1000 1 1 60 Barrier_sym 120 100 90 30 30 5  25.0788
BlackSholes	120 100 (nothing)6 30 5 Call=24.458	bs 120 100 6 30 5  24.458

## 4 計算結果の記録と保存

計算結果をメモするのも大変なので必要なもののみ保存できるようにする。ファイルシステムは *Jfiles* を用いた。

\*4

*Jfiles* はリスト (ベクトル) やマトリクスが自然に取り扱うことが出来る。結果は暗号化されており堅固であるが、エディタで見るとは出来ない。

*Jfiles* の詳細は別稿とした。

- J602 のフォルダに対応した。

\*4 EXCEL を毎回リンクさせ、ベクトルをスカラに直して EXCEL に落とす方法もあるが面倒である

```
jcreate jpath '~temp/barrierx' x is 0 1 2..
```

- ファイルネームは *barrier0.ijf, barrier1.ijf*
- extension は J が自動で付ける。(ijf) ファイルを呼び出すときも自動識別するので ijf は記述しなくても良い。
- New/Reset 初期化。同じファイルネームだと上書きで消される。
- ファイルネームは *barrierx(ijf) x is 0 1 2..*
- 書き込みは append のみ。ボタンをクリックすると書き込まれる。
- ファイルとレコードの確認。 *ijx* での操作。0 から 4 レコードある。

```
jsize jpath '~temp/barrier0'  
0 4 2304 0
```

- 読み出し
  - *ijx* での操作
  - jpath 以下を括弧で囲む。
  - 2重ボックスを用いているので開く (>)
  - X Y ANS である

```
> jread (jpath '~temp/barrier0');0 1 2 3  
+---+-----+-----+  
|1 1|120 90 100 6 30 5|27.9592|  
+---+-----+-----+  
|1 1|120 90 100 6 40 5|26.1784|  
+---+-----+-----+  
|1 1|120 90 100 6 50 5|24.9333|  
+---+-----+-----+  
|0 |120 90 6 30 5 |32.8962|  
+---+-----+-----+
```

## 5 Reference

竹内寿一郎「エキゾチックオプション価格のシュミレーション(その11)バリアオプションの価格式(6)」  
JAPLA2009/04

[http://homepage3.nifty.com/assgaya\\_avenue](http://homepage3.nifty.com/assgaya_avenue)